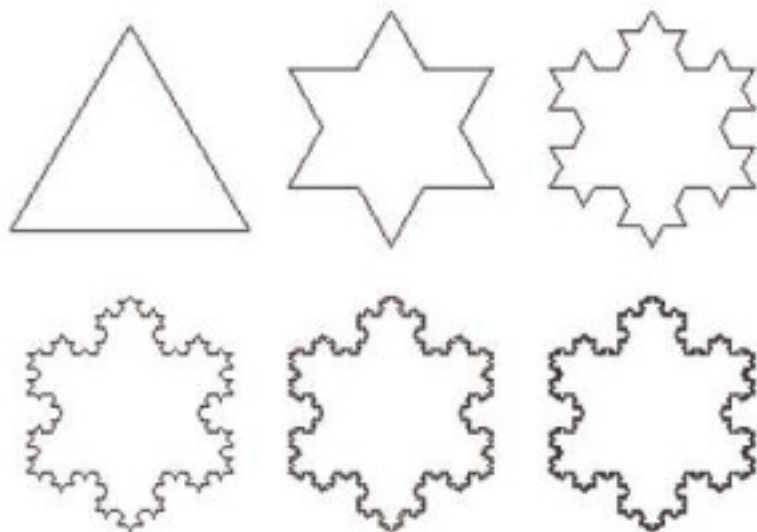
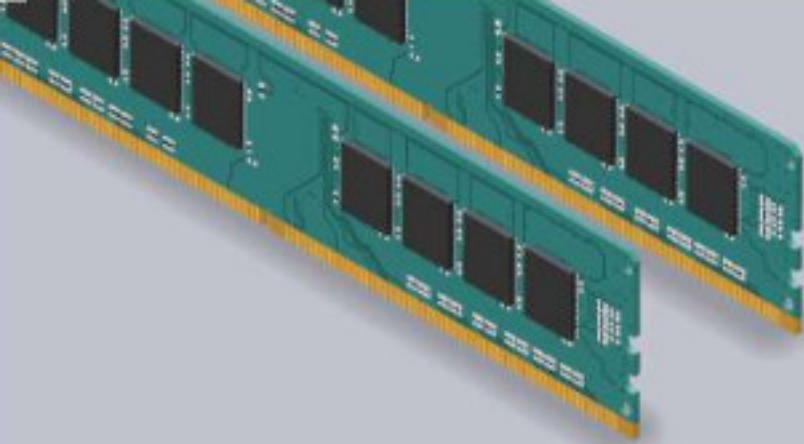
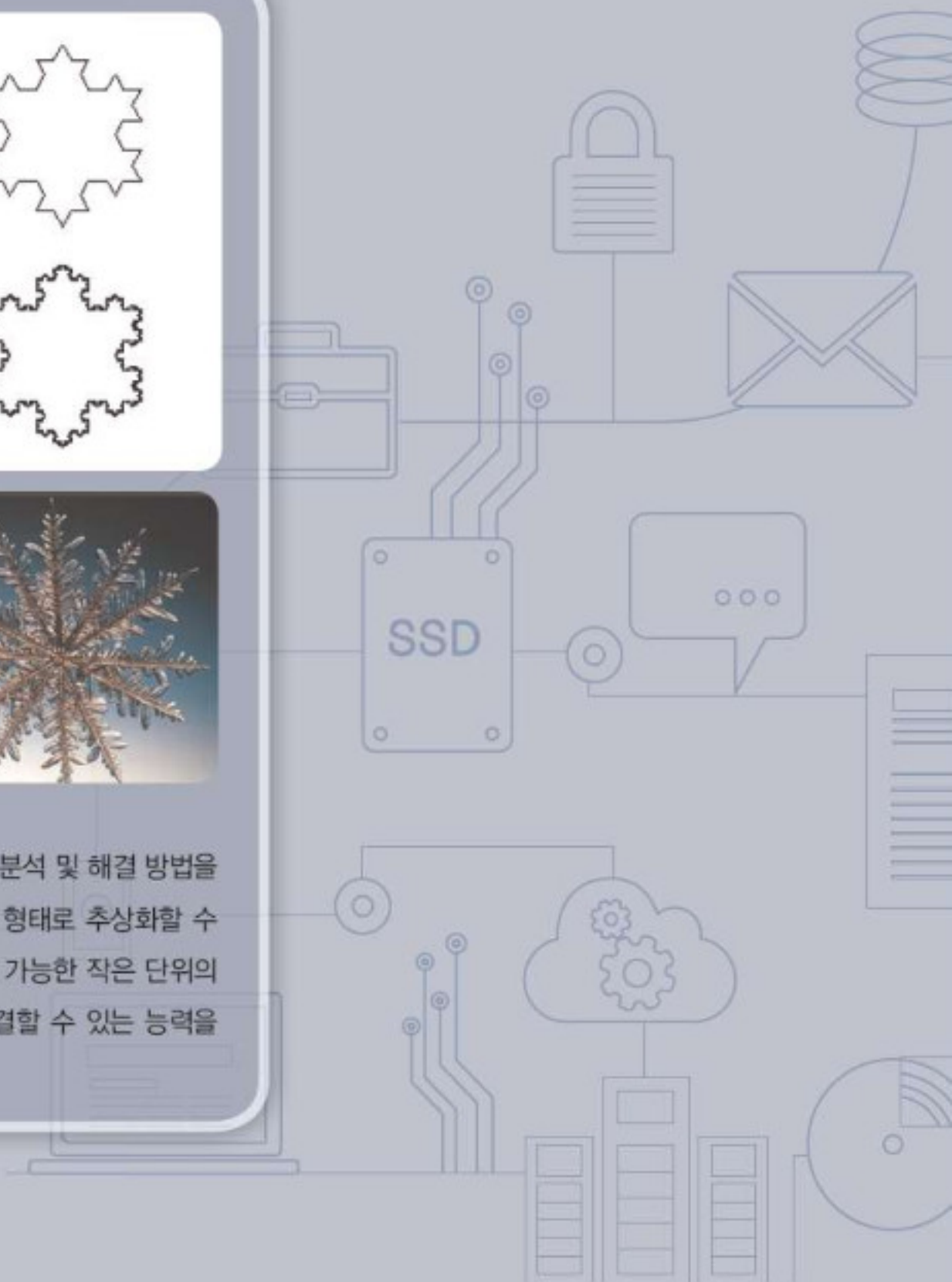




우리는 생활하면서 여러 가지 문제 상황에 부딪치게 된다. 이러한 문제 상황들을 해결하는 데 도움을 주는 프로그램들은 정보 과학적인 문제 해결 절차를 적용하여 만들어진 것으로, 컴퓨터의 정확하고 신속한 처리 능력과 합해져 유용하게 활용되고 있다.



이 단원에서는 문제를 이해하고 직업 내용 분석 및 해결 방법을 찾아 필요한 자료를 추출하여 처리에 적합한 형태로 추상화할 수 있는 역량을 기르며, 문제를 독립적으로 처리 가능한 작은 단위의 문제로 나누어 연관된 문제를 논리적으로 연결할 수 있는 능력을 기를 수 있도록 한다.



PART

V

문제 해결과 알고리즘

> CHAPTER <

01. 추상화

02. 알고리즘 설계



학습 목표

- 문제를 해결하는 과정을 설명할 수 있다.
- 문제를 이해하고 작업 내용 분석 및 해결 방법을 찾아 필요한 자료를 추출할 수 있다.
- 문제 해결을 위한 처리에 적합한 형태로 추상화할 수 있다.
- 문제를 독립적으로 처리 가능한 작은 단위의 문제로 나누어 연관된 문제를 논리적으로 연결할 수 있다.

여는 이야기

일상생활에서 문제 해결

일상생활 속에서 우리는 다양한 문제를 접할 수 있다. 그리고 어떤 경로를 선택하는가에 따라 시간과 비용이 달라지는 경우를 볼 수 있다. 우리의 생활 속에서 접하는 문제를 해결하는 과정을 생각해 보자.



단/원/학/습/안/내

이 단원에서는 우리 생활과 산업 환경에서 발생하는 문제를 해결하는 원리와 과정을 살펴보고, 문제를 해결하기 위한 추상화가 무엇인지 살펴봄으로써 실생활과 산업 현장에서 관련된 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1

문제 해결 과정

문제는 해결해야 할 과제나 상황을 의미하며, 문제를 해결한다는 것은 원하는 결과를 얻어낸다는 것을 의미한다. 예를 들어 어떻게 하면 더 빠르게 목적지에 갈 수 있을까? 어떻게 하면 일을 더 빨리 할 수 있을까? 등은 일상생활에서 쉽게 접할 수 있는 문제이다.

(1) 문제 해결 과정

문제가 발생하면 해결할 수 있는 다양한 방법을 생각하게 된다. 어떤 방법을 선택하는가에 따라 결과는 달라질 수 있다. 그러므로 문제를 이해하고 분석하여 최적의 해결 방법을 선택하고 실행하는 과정이 필요하다. 이와 같이 효율적으로 문제를 해결하기 위한 과정을 문제 해결 과정이라고 한다.

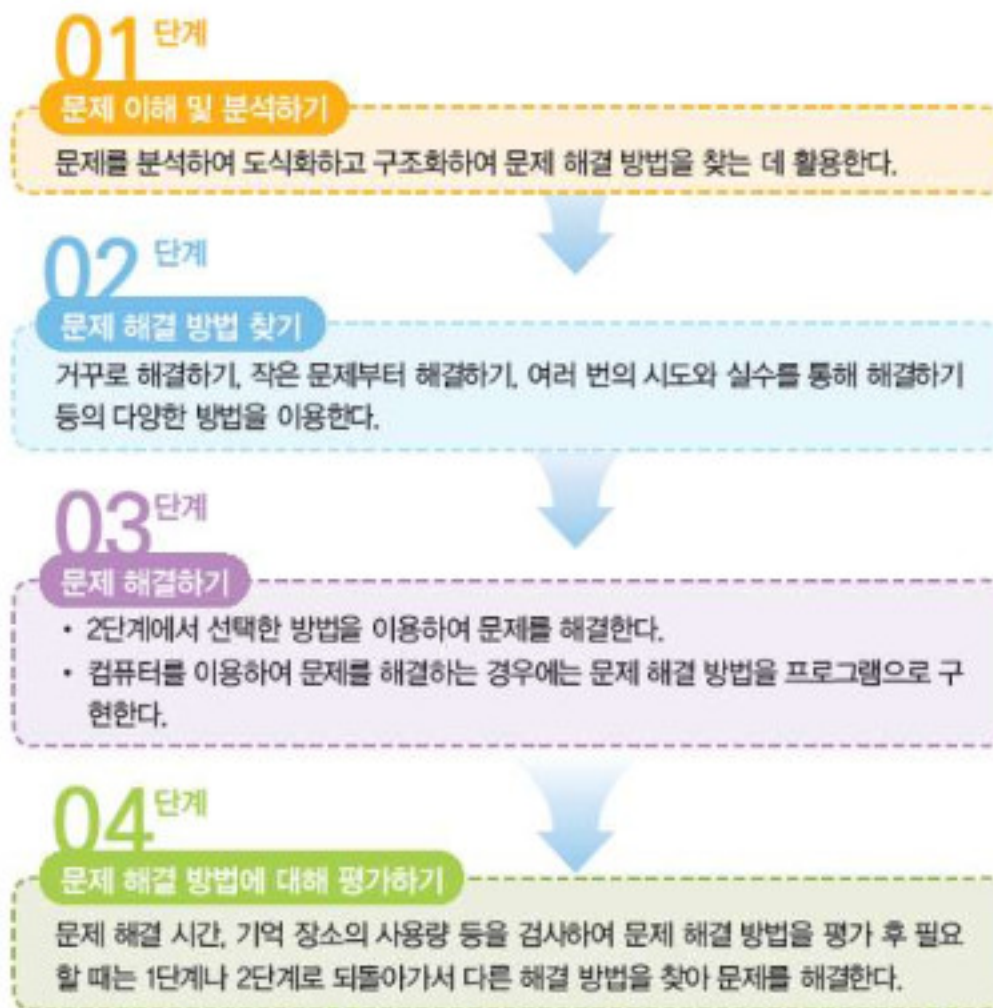


그림 V-1 문제 해결 과정과 생활 속 사례

(2) 문제의 이해 및 분석

문제를 효율적으로 해결하기 위해 문제를 이해하고 정확하게 분석하는 과정이 필요하다. 문제 이해를 통해 문제 상황을 이해하고 문제 해결에 필요한 요소와 필요하지 않은 요소를 구분한다. 문제 분석을 통해 문제를 해결하는 데 필요한 정보를 수집하고, 수집한 정보를 적합한 방법으로 표현한다.



작업 따라잡기

문제를 이해하고 분석하기

문제를 파악하고 문제를 해결하기 위해 필요한 정보는 무엇인지 알아보자.

농구 동아리 회장인 진욱이가 학급 대항 농구 대회를 개최하려고 한다. 일주일 후는 방학이어서 그전까지는 모든 경기를 마쳐야 하는데 어떻게 해야 할지 고민이다. 단, 진욱이네 학교는 총 학급수가 16개이다.

▶ 문제 이해하기

1. 진욱이의 고민은 무엇인가?

2. 참가팀 수에 따라 경기 수가 어떻게 달라지는지 살펴보자.

▶ 문제 분석하기

이 문제를 해결하기 위해 필요한 정보는 다음과 같다.

경기 방식

경기에 참여하는 팀 수

다음 토너먼트 방식의 대진표를 작성하고, 팀의 수에 따른 경기 수를 파악해 보자.

팀 수	경기 수
2	
4	
8	
16	

(3) 문제 표현 방법

문제 분석한 결과를 명확하고 구체적으로 표현하는 것이 필요하다. 문제를 표현함으로써 문제를 쉽게 이해할 수 있고 문제를 분석하고 해결하는 데 도움이 된다.

1 글로 표현하기

머릿속에 있는 문제를 글로 표현해 보는 것이다. 이때 정리하여 적을 필요는 없으며 생각나는 대로 적어보면 문제를 이해하고 해결해 나가는 데 도움이 된다.



예

- 이번 주 해야 할 일
1. 수학문제 풀기
 2. 독후감 2편 작성하기
 3. 팀별 발표 프젠테이션 작성하기

그림 V-2 말과 글로 표현하기

2 표로 표현하기

핵심적인 내용을 일목요연하게 정리하여 표의 형태로 표현한다. 자료의 형태에 따라서는 표만으로 많은 자료를 요약하거나 각 변수와의 관계, 수치 분포 모양, 또는 시간에 따른 추이 등을 명료하게 나타내기 어려울 때가 있다. 이러한 경우에는 그래프, 차트 등의 기법을 이용하면 시각적인 이해와 판단에 많은 도움이 된다.

방법1		
출발	도착	거리(m)
집	버스정류장	25
버스정류장	문구점	30
문구점	학교	25
총 거리		80

그림 V-3 표로 표현하기

3 그림과 기호로 표현하기

약속된 기호나 도형을 이용하여 제시된 문제를 시각화해보면 복잡한 문제를 간단하게 표현할 수 있다. 하지만 사용하는 기호 등이 사전에 약속되어 있지 않으면 다른 사람이 이해하기 어려울 수 있다.

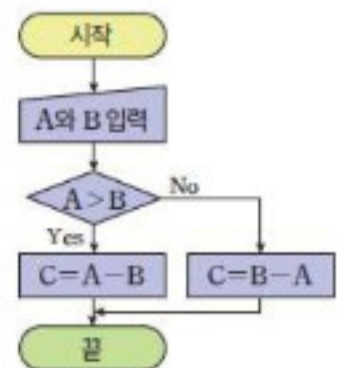


그림 V-4 그림, 기호, 도형으로 표현하기

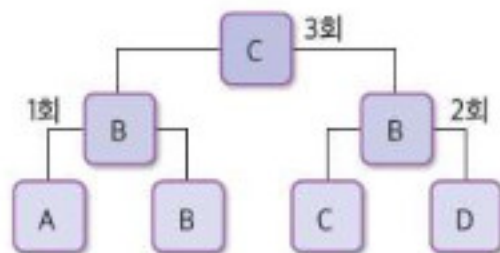


경기 수를 파악하기 위한 문제를 표현해 보자.

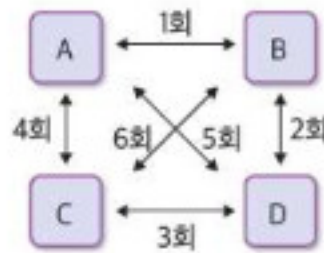
▶ 글로 표현하기

2팀일 경우에는 1경기, 4팀일 경우에는 3경기, 8팀일 경우에는 7경기

▶ 그림으로 표현하기



토너먼트 방식



리그방식

▶ 표로 표현하기

팀 수	경기 수
2	
4	
8	
16	

(4) 컴퓨터와 문제 해결

해결할 문제의 양이 많거나 자동화 처리가 필요한 경우 컴퓨터로 문제를 해결하여 효율을 높일 수 있다. 컴퓨터로 문제를 해결하고자 할 때 과정은 다음과 같다.

1 문제 이해 및 분석

해결하려고 하는 것이 무엇인지 찾기 위해 제시된 문제를 잘 이해하고 분석하는 단계로, 일반적인 문제 해결 과정과 동일하다.

2 문제 해결 방법 찾기

일반적인 문제 해결 과정의 단계와 동일하게 주어진 문제를 해결하기 위한 가장 효율적인 방법을 찾아본다.

3 알고리즘 설계

알고리즘은 문제 해결 과정에서 규칙성을 찾아 논리적인 순서로 설명하거나 표현하는 문제 해결 절차나 방법을 말한다. 따라서 알고리즘 설계는 이러한 문제 해결 절차나 방법을 알기 쉽게 기술하는 것이다.

컴퓨터로 문제를 해결하려고 할 때 알고리즘을 표현하는 방법으로는 자연어, 의사코드, 순서도, 프로그래밍 언어 표현을 사용한다.

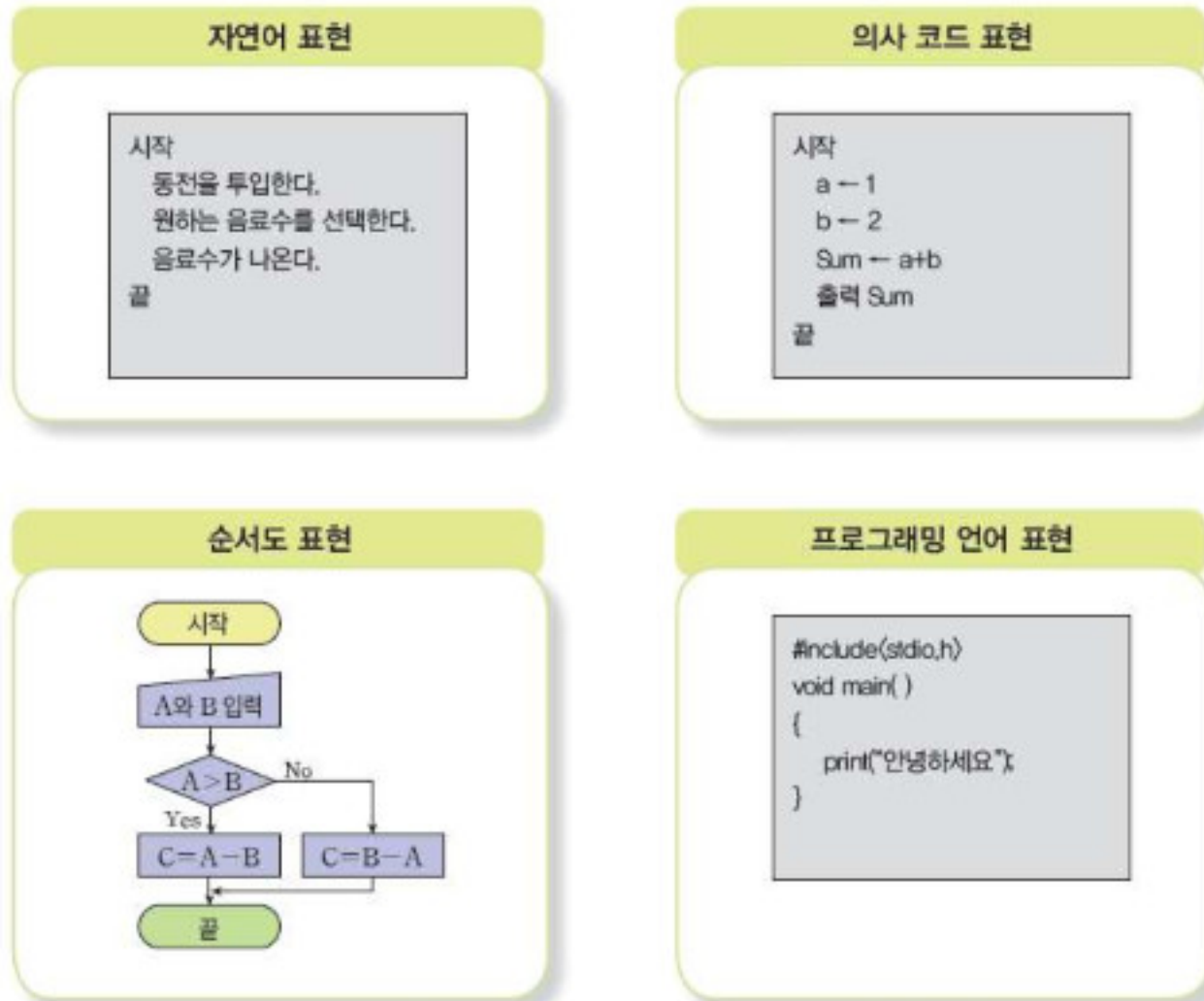


그림 V-5 문제 표현

4 프로그램 작성

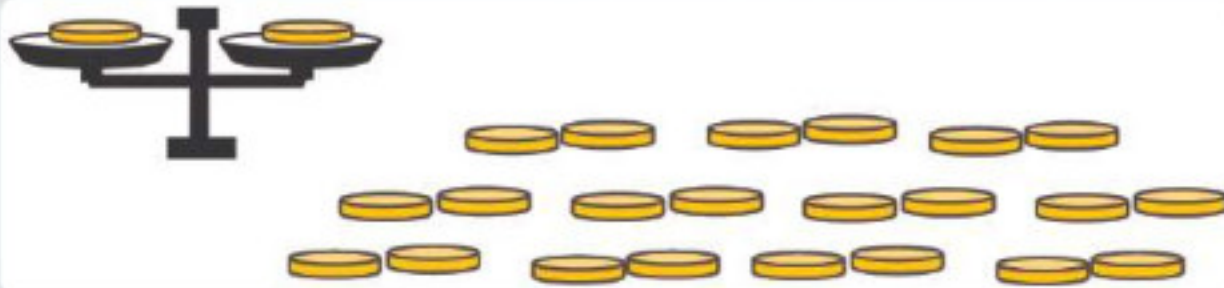
컴퓨터를 이용한 문제 해결 과정에서는 알고리즘 설계를 바탕으로 프로그래밍 언어를 이용하여 프로그램을 작성해야 하는 부분이 있는지 확인한다. 프로그램이 제대로 실행되는지 확인하여 만약 프로그램 실행 결과가 잘못되었거나 오류가 발생한다면 다시 수정해야 한다.



문제

다음의 가짜 동전 문제를 해결하시오.

아주 많은 동전 더미 속에 1개의 가짜 동전이 섞여 있다.
가짜 동전은 매우 정교하게 만들어져 누구도 눈으로 식별할 수 없다.
그러나 가짜 동전의 무게는 정상적인 동전보다 약간 가볍다.
가짜 동전 찾기 문제는 이 가짜 동전을 찾아내기 위해서 양팔 저울만 사용하여 가짜 동전을 찾아내는 것인데,
가능한 한 저울에 동전을 다는 횟수를 줄여야 한다.



질문

1. 문제를 해결하기 위해 문제를 분석하고 문제를 정의해 보자.

2. 문제를 해결할 수 있는 다양한 방법을 찾아보자.

3. 위의 방법들 중 하나를 선택하고, 그 이유를 찾아보자.

4. 선택한 방법을 수행하는 과정을 표로 표현해 보자.

(1) 추상화

문제를 해결하기 위해서는 필요한 자료를 수집 및 분석하고, 필요한 표현 방법(도표, 그래프 등)을 활용하여 눈으로 보기 쉽게 나타내고, 복잡한 요소를 작은 단위로 분해하고, 해결에 필요한 변수들을 추출하여 적절한 해결 모델을 설계하는 과정이 필요하다.

추상화는 문제 해결을 위해 문제를 분해하거나 중요한 부분을 끌어내는 것 등을 통하여 해결해야 할 문제를 단순화하고, 쉬운 형태로 변환하는 과정이다. 연속적인 데이터를 이산적인 데이터로 표현하는 과정, 실제 현상을 절차화된 언어/숫자/수식으로 변환하는 과정을 예로 들 수 있다.

문제의 이해 및 분석
과정에서의 추상화

- 문제 속에 포함된 불필요한 요소들을 제거
- 필요한 부분을 추출

1 문제의 분해(problem decomposition)

문제의 분해는 주어진 문제를 여러 개의 작은 문제들의 집합으로 볼 수 있는 경우에 적용하는 문제 이해 및 분석 방법이다. 크고 복잡한 문제보다는 작고 단순한 문제를 해결하기가 쉽다.

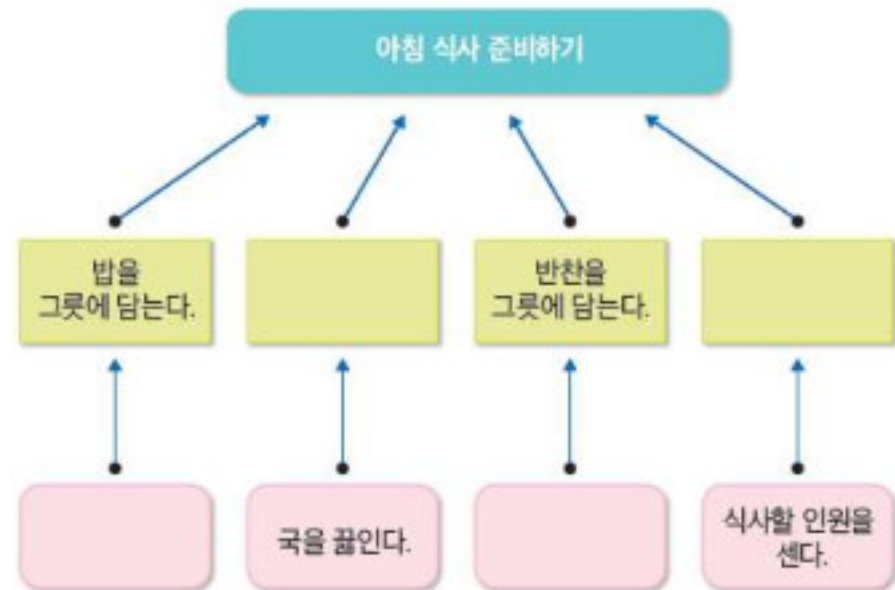


그림 V-6 문제의 분해(예: 아침 식사 준비하기)

직무 따라잡기

분석을 통한 추상화 과정을 이해해 보자.

(1) 리그 방식으로 축구 경기를 운영할 경우 경기 수를 알아 보자.

팀 수	경기 수
2	1
3	3
4	6
5	10

(2) 팀 수와 경기 수 간 어떤 규칙이 있는지 찾아보자.

팀 수	규칙	경기 수
2	1	1
3	1+2	3
4	1+2+3	6
5	1+2+3+4	10
n	1+2+3+4+...+(n-1)	

2 모델링(modeling)

모델링이란 문제 해결을 위해 문제 내부에 포함된 구성 요소들 간의 관계를 단순화해 표현하고 구조화하는 것이다. 예를 들어, 3차원 CAD에서는 제품 등의 입체 형상을 컴퓨터 내부의 가상 공간에 만들어 내는 것도 모델링이라고 할 수 있다. 모델링을 통해 만들어지는 새로운 형식 또는 구조가 바로 모델이다.



작업 따라잡기

모델링을 통한 추상화 과정을 이해해 보자.

(1) 아래의 그림에서 (A)와 (B) 모델 중 문제 해결 방법을 설계하는 것이 쉬운 것은?

(2) 3학년 드림반의 체육 부장은 누구인가?

(A) 우리 2학년 3반은 반장인 김은정과 부반장인 이하나. 그 아래 학생부, 총무부, 체육부, 학습부, 환경 관리부의 다섯 부서가 있다. 각 부서의 부장은 정석원, 박선아, 최효재, 이성준, 강재민이며 각 부서별 부원이 4명씩 있다.



그림 V-7 추상화를 통한 모델링(예: 학급 구성도)



앨런 튜링은 정보 사회를 형성하는 근원적인 생각을 처음으로 제시했던 사람이었지만, 많은 사람들이 그에 대해 거의 알지 못합니다. 컴퓨터가 만들어지기 전에 그는 숫자 계산을 위한 추상화 도구인 튜링 기계라는 이론적인 기계의 구조와 동작 원리를 고안하였습니다. 구글에서는 그의 탄생 100주년을 기념하기 위해 튜링 기계의 실제적인 동작을 보여주는 12개의 프로그래밍 퍼즐(1~6번까지의 퍼즐은 7~12번까지의 퍼즐을 해결하기 위한 힌트입니다.)을 제시하고 있습니다

튜링의 업적은 단순히 튜링 기계로 한정지을 수 없을 만큼 엄청납니다. 그는 암호 코드를 해독하여 제2차 세계 대전을 단시간에 종식시켰으며, 초창기 컴퓨터 기술의 개척자였습니다. 그는 인공지능 분야의 선구자이며, 이른바 튜링 테스트를 통해 컴퓨터가 생각하는 기계임을 증명하기도 하였습니다. 그의 업적은 추상적인 수준이 아니었습니다. 그는 실제 기계를 설계하고 개발하였는데, 이 기계 속에 그의 생각을 표현하고 회로로 구현하였습니다. 이러한 순수 수학과 컴퓨팅 기계의 결합을 통해 오늘날 정보 과학의 초석을 다졌습니다. 그가 남긴 업적들은 현대 컴퓨팅 기술 내부에 살아 숨 쉬고 있습니다.

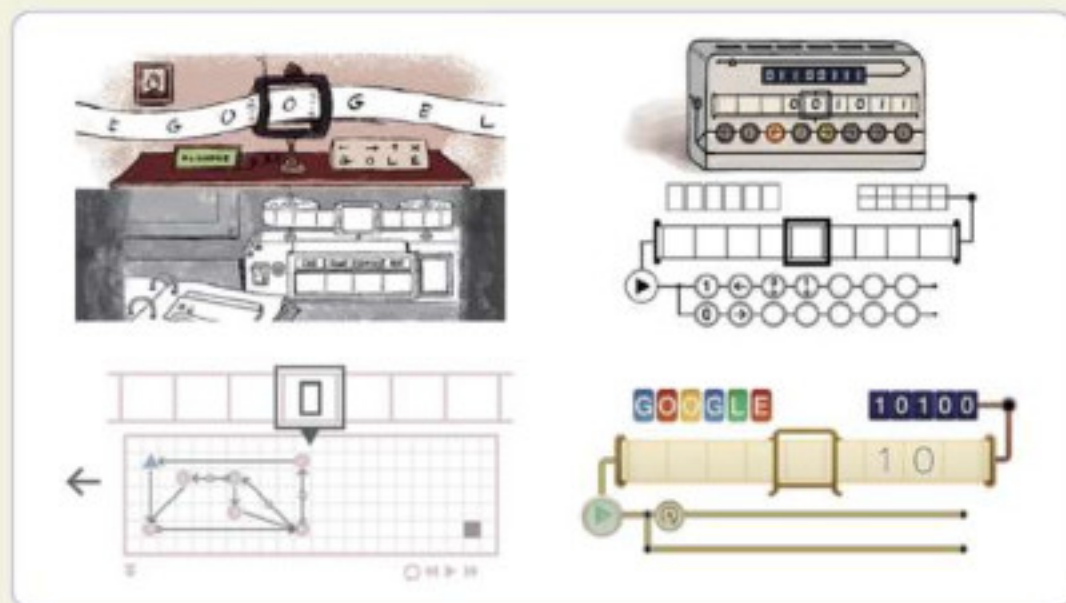


그림 V-8 구글에서 설계한 다양한 형태의 튜링 기계

튜링의 100주년을 기념하기 위해 아주 특별한 두들을 개발하고 싶었습니다. 그래서 그의 업적에 관한 깊이 고찰하였으나, 대부분이 추상적이거나 잘 알려지지 않아서 실제로 표현하기 어려웠습니다. 그래서 가시적으로 동작하는 모습을 구현하기 위한 다양한 설계들을 해보았습니다. 튜링 머신은 이론적으로 존재하는 논리적 기계이기 때문에 기술적인 정확성과 구현 가능성을 고려해야만 했습니다. 그리고 이를 표현할 수 있는 가시적인 디자인과 대략적인 난이도를 고려하여 퍼즐을 선별하였습니다. 따라서 기존의 두들에 비해 훨씬 더 많은 테스트를 할 수밖에 없었습니다. 우리의 영웅인 튜링에게 바치는 본 작품을 여러분들이 즐길 수 있기를 바랍니다.

- 출처 : 앨런 튜링 탄생 100주년 기념(2012.06.23). 구글.
(<https://www.google.com/doodles/alan-turings-100th-birthday>)



활동 주제

시설 안전 점검을 위해 병아리 유치원을 방문하려고 한다. 최대한 짧은 시간에 검사를 마치려면 어떤 경로로 점검을 해야 할까?



● 이동 조건 ●

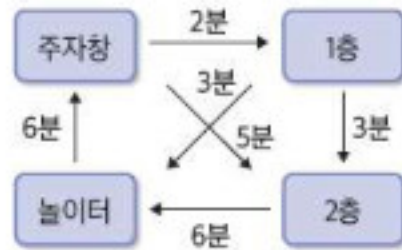
- 주차장에서 시작해서 주차장으로 돌아와야 한다.
- 주차장에서 현관까지 이동 시간은 2분이다.
- 1층에서 2층까지 이동 시간은 3분이다.
- 놀이터에서 현관까지 이동 시간은 3분이다.
- 주차장에서 놀이터까지 이동 시간은 6분이다.



이동 구간	이동 시간
주차장-현관	2
1층-2층	3
놀이터-현관	3
주차장-놀이터	6

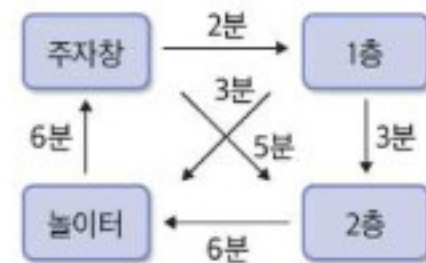
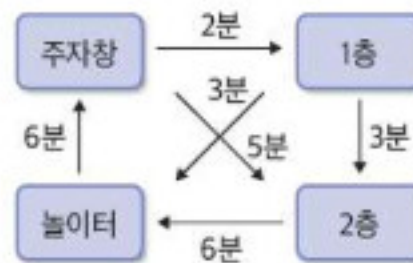
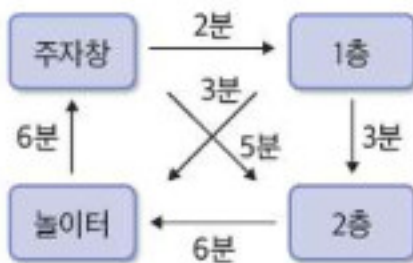
▶ 점검 구역 확인하기 : 주차장, 1층, 2층, 놀이터

▶ 구역 간 소요 시간 확인하기



▶ 경로 찾기

방법	경로	시간
1	주차장-1층-2층-놀이터-주차장	
2	주차장-2층-놀이터-1층-주차장	
3	주차장-놀이터-1층-2층-주차장	



알고리즘 설계

학 습 목 표

- 알고리즘의 개념과 표현 방법을 이해하고 자연어, 의사코드, 순서도를 이용하여 알고리즘을 표현할 수 있다.
- 순차, 선택, 반복 구조 유형으로 알고리즘을 설계할 수 있다.
- 다양한 문제를 순서도 기호를 사용하여 알고리즘을 표현할 수 있다.

여는 이야기

알파고는 어떻게 작동할까?

인공지능 프로그램 AlphaGo가 세계 챔피언 이세돌 프로9단을 꺾었다.

'어떻게 컴퓨터는 사람처럼 바둑을 두는 것이 가능할까?' 알파고의 인공지능도 알고리즘을 바탕으로 작동한다. 알고리즘의 개념을 알고 설계 및 표현방법을 생각해보자.



단/원/학/습/안/내

이 단원에서는 알고리즘의 설계 과정을 살펴보고, 다양한 문제를 알고리즘의 구조와 표현 방법으로 나타냄으로써 실생활과 산업 현장에서 관련된 직무를 수행하는 데 필요한 실무 능력을 기르도록 한다.

1

알고리즘의 개념

우리는 주어진 문제를 보다 효율적으로 해결하기 위해 여러 가지 해결 방법과 단계별 순서를 생각해 보고, 그중 가장 적합한 방법을 선택하여 실행한다. 이와 같이 어떤 문제를 해결하기 위한 절차나 방법을 알고리즘이라고 한다. 예를 들어 십진수를 이진수로 바꾸는 방법, 인터넷에서 책을 구입하는 방법, 라면을 끓이는 조리법 등은 모두 알고리즘이다.

조리방법



그림 V-9 알고리즘의 예

(1) 알고리즘의 필요성

컴퓨터는 사람과 달리 스스로 문제를 이해하고 분석할 수 없기 때문에 어떤 동작을 어떤 순서로 처리해야 하는지 단계적으로 명확하게 지시해야 한다. 그러므로 컴퓨터를 이용하여 문제를 해결하고자 할 때는 논리적이고 체계적으로 정리된 알고리즘이 필요하다.

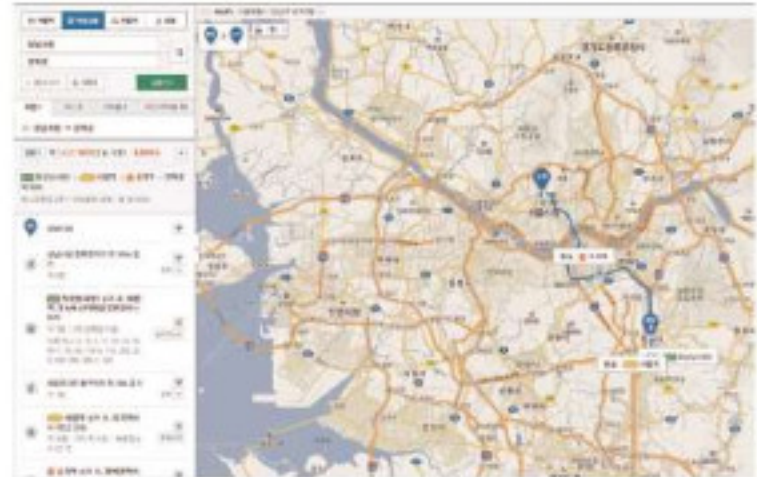


그림 V-10 알고리즘의 활용

직무 따라잡기

알맞은 순서대로 나열해 보자.

(1) 인터넷에서 책을 구입하는 방법

- ① 인터넷 서점 사이트에 접속한다.
 - ↓
 - ② 검색한 책을 주문한다.
 - ↓
 - ③ 필요한 책을 검색한다.
 - ↓
 - ④ 주문 사항을 확인하고 결제한다.
- () → () → () → ()

(2) 십진수를 이진수로 바꾸는 알고리즘

- ① 십진수를 2로 나눈다.
 - ↓
 - ② 나머지를 역순으로 나열한다.
 - ↓
 - ③ 몫이 2 이상이면 계속 나눈다.
- () → () → () → ()

(2) 알고리즘의 조건

문제에 대한 해결 과정이나 방법을 설명한다고 해서 모두 알고리즘이라고 할 수는 없다. 정확하지 않거나 논리적이지 못한 방법으로는 문제를 해결할 수 없기 때문이다. 알고리즘은 다음과 같은 조건을 모두 갖추어야 한다.

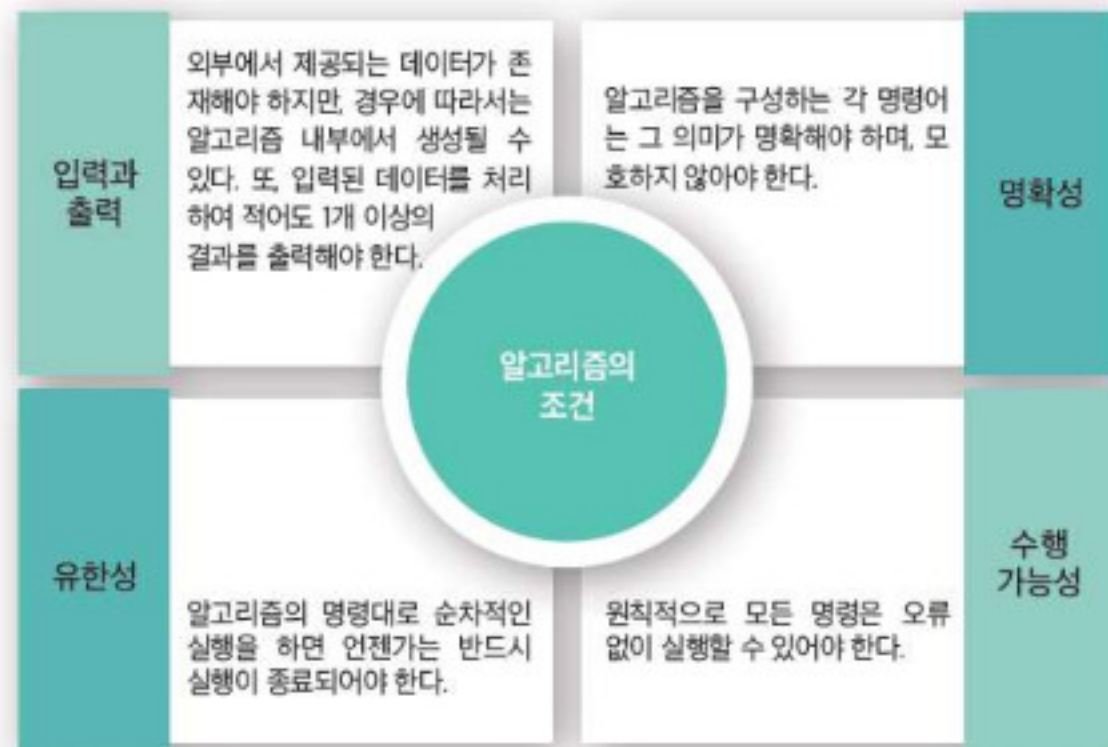


그림 V-11 알고리즘의 조건

작업 따라잡기

▶ 1에서 n 까지의 수 중에서 홀수의 합을 구하는 알고리즘이다. 알고리즘이 갖추어야 할 조건을 모두 갖추고 있는지 확인해 보자.

- ① m 을 1로 초기화한다.
- ② sum 은 0으로 초기화한다.
- ③ 만약 m 이 n 보다 크면 ⑦로 이동한다.
- ④ 그렇지 않으면 sum 에 m 을 누적한다.
- ⑤ m 의 값을 2 증가하고 ④로 이동한다.
- ⑥ sum 의 값을 출력하고 작업을 종료한다.

조건	만족 유무 및 이유
입력 및 출력	
명확성	
유한성	
수행 가능성	

(3) 알고리즘의 표현 방법

문제 해결 과정을 설명한 알고리즘은 다양한 형태로 표현할 수 있다. 일반적으로 알고리즘을 표현하는 방법에는 자연어, 순서도, 의사 코드, 프로그래밍 언어 등이 있다.

1 자연어(Natural Language) 표현 방법

일상생활에서 사용하는 언어를 이용하여 알고리즘을 표현하는 방법이다. 별다른 지식이 없어도 쉽게 사용할 수 있어 편리하지만 일상 언어의 모호함 때문에 읽는 사람에 따라 내용이 다르게 해석될 수도 있으므로 최대한 명료하게 표현해야 한다.

숫자를 입력받아 홀수, 짝수를 판별하는 알고리즘을 자연어로 표현하면 다음과 같다.

- ① 숫자를 입력받는다.
- ② 입력된 숫자를 2로 나눈다.
- ③ 나머지가 0이면 짝수를 출력한다.
- ④ 나머지가 1이면 홀수를 출력한다.

2 순서도(Flowchart) 표현 방법

미리 약속된 기호를 사용하여 순서대로 일관성 있게 알고리즘을 표현하는 방법이다. 전체적인 흐름을 명확하게 이해할 수 있고 오류를 쉽게 찾을 수 있다는 장점이 있지만, 알고리즘의 규모가 큰 경우에는 오히려 복잡할 수 있다.

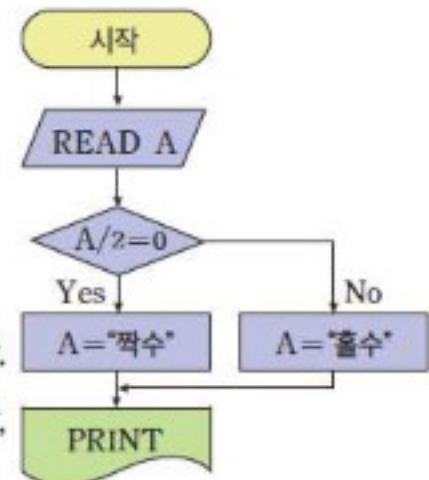


그림 V-12 순서도의 예

3 의사 코드(Pseudo-code) 표현 방법

일반적인 언어를 프로그래밍 언어처럼 나열하는 방법으로 자연어와 프로그래밍 언어의 중간에 해당한다. 특정 프로그래밍 언어에 대한 지식이 없어도 자유롭게 알기 쉬운 표현이 가능하고, 다른 프로그래밍 언어로 전환하기가 쉽다.

숫자를 입력받아 홀수, 짝수를 판별하는 알고리즘을 자연어로 표현하면 다음과 같다.

```

BEGIN
  a ← 2;
  b ← a%2;
  if b==0 PRINT "짝수";
  if b==1 PRINT "홀수";
END
  
```

그림 V-13 의사 코드 표현 방법의 예

```

#include <stdio.h>
int main(void){
  int num1 = 0;
  int num2 = 0;
  printf("enter number");
  scanf("%d", &num1);
  num2 = num1%2;
  if(num2==0){
    printf("짝수입니다.");
  }
  else{ printf("홀수입니다.");
  }
  return 0;
}
  
```

그림 V-14 프로그래밍 언어 표현의 예

4 프로그래밍 언어(Programming Language) 표현 방법

컴퓨터 프로그래밍 언어를 사용하여 알고리즘을 표현하는 방법이다. 컴퓨터를 이용하여 바로 실행할 수 있어 편리한 반면 사용자가 복잡한 프로그래밍 문법을 이해하고 있어야 하는 단점이 있다.

알고리즘을 의사 코드와 순서도로 표현해 보자.

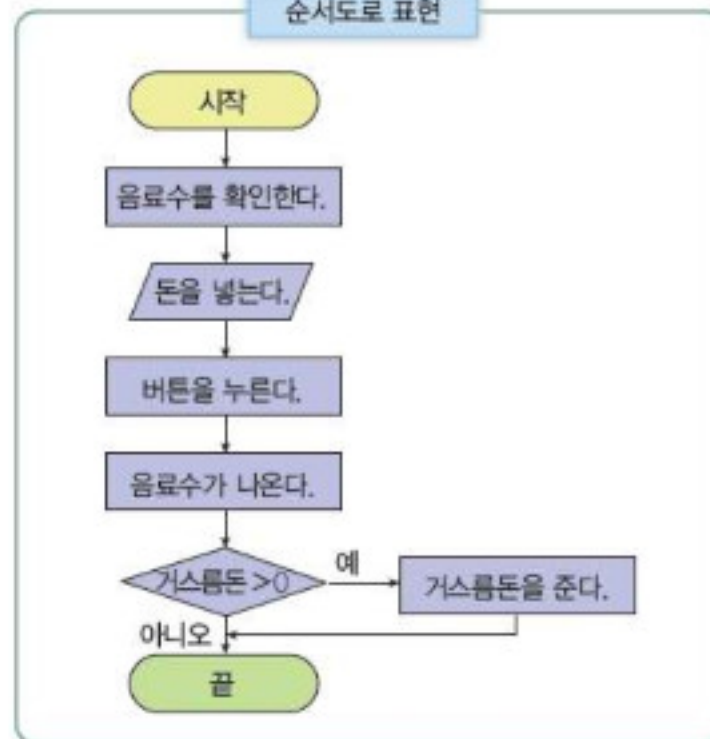
자동판매기에서 동전을 넣어 음료수를 선택해 나오기까지의 과정을 알고리즘으로 표현해 보자.



자연어로 표현

- ① 원하는 음료수가 있는지 확인한다.
- ② 돈을 넣는다.
- ③ 원하는 음료수가 표시된 버튼을 누른다
- ④ 음료수가 나온다
- ⑤ 거스름돈을 주어야 하는가?
"예"이면 ⑥번으로 이동한다.
"아니오"면 ⑦번으로 이동한다.
- ⑥ 거스름돈을 준다.
- ⑦ 종료한다.

순서도로 표현



(4) 알고리즘의 설계 구조

알고리즘 설계는 문제 해결 과정을 규칙적으로 만들기 위해 구체적인 방법을 기술하는 것이다. 알고리즘 설계를 통해 문제를 정확하게 분석하고, 논리적이고 효율적인 최적의 문제 해결 방법을 찾을 수 있다.

알고리즘 설계 시 유의사항은 다음과 같다.

- 문제를 이해하여 분석한다.
- 분석한 문제를 기호, 그림 등으로 표현한다.
- 문제의 표현이 해결하기에 적당한지 검토한다.
- 알고리즘이 논리적이며 다섯 가지 조건 (입출력, 명확성, 유한성, 수행 가능성)을 갖추었는지 분석한다.
- 알고리즘에 오류가 있는지 살펴보고 수정한다.
- 수정된 알고리즘을 프로그래밍하기에 적합하게 표현한다.

1 순차 구조

시작부터 끝까지 차례대로 명령을 실행하는 구조이다. 조건에 따라 명령이 바뀌거나 특정 부분을 반복하여 실행하지 않고, 명령문을 단순히 위에서 아래로 단계적으로 처리하는 경우에 사용한다.



그림 V-15 순차 구조

2 선택 구조

주어진 조건에 따라 처리 내용이나 순서가 달라지는 구조이다. 조건을 만족하면 “예 (True/Yes/만족)”인 경우의 실행 결과를 처리하고, 조건을 만족하지 않으면 “아니요 (False/No/불만족)”인 경우의 실행 결과를 처리한다.

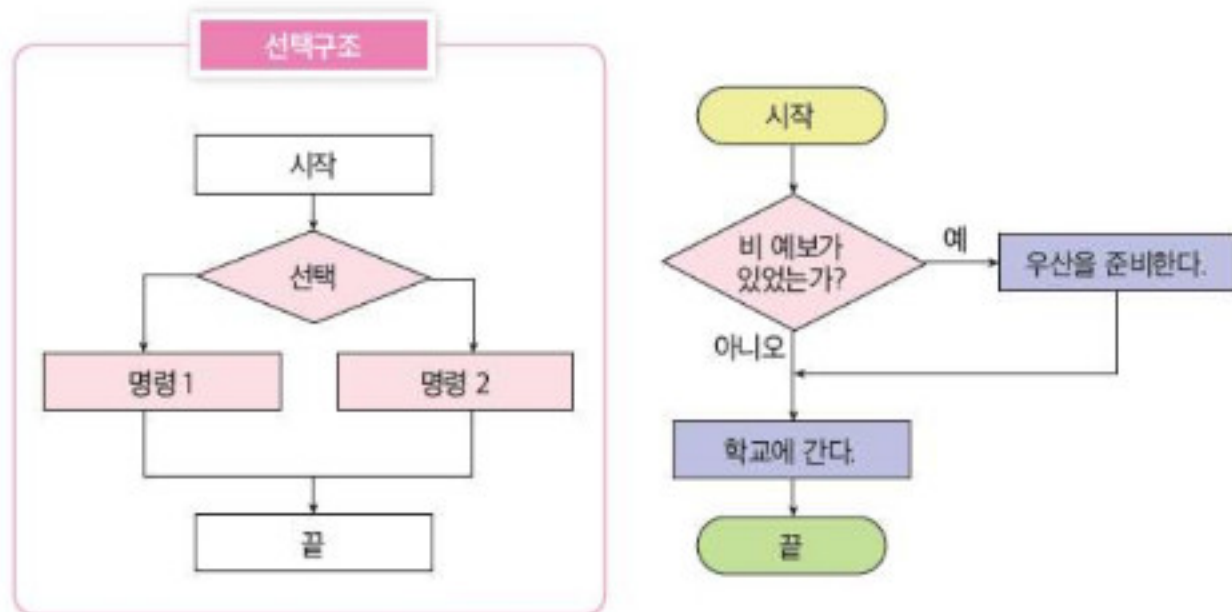


그림 V-16 선택 구조

알고리즘 분석 기준

- 정확성
실행 결과가 오류없이 정확한가
- 단순성
알고리즘의 구조나 흐름이 이해하기 쉬운가
- 수행시간 비교
알고리즘을 수행하는 데 걸리는 시간
- 기억 장소 사용량
알고리즘을 수행하는데 차지하는 기억 장소의 크기

3 반복 구조

주어진 조건의 만족 여부에 따라 일련의 명령문을 반복 수행할 때 작성한다.

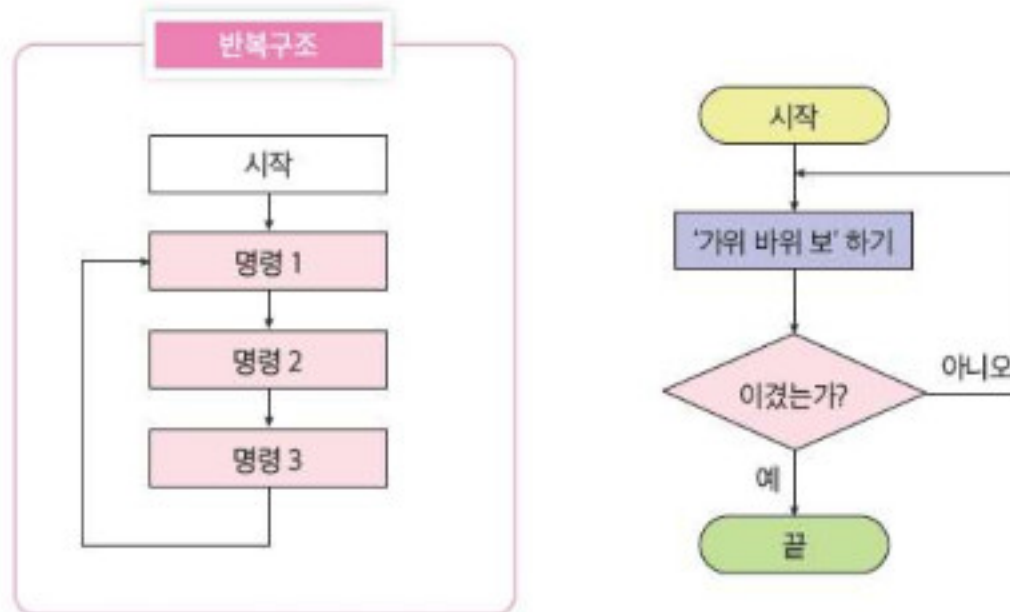


그림 V-17 반복 구조

작업 따라잡기

우리 주변에서 순차, 선택, 반복 구조로 설계할 수 있는 문제를 찾아보자.

설계 구조	문제 상황
순차	
선택	
반복	

2 순서도를 이용한 문제 해결

문제를 해결하기 위해 순서도를 사용할 때에는 다음과 같이 약속된 기호를 사용한다.

표 V-1 순서도의 기호

이름	기호	설명
단말기		시작과 끝 순서도의 시작과 끝을 표시할 때 사용한다.
준비		준비 변수나 조건의 선언 및 데이터 초깃값을 설정할 때 사용한다.
입출력		데이터 입출력 자료의 일반적인 입력 및 출력에 사용한다.
처리		데이터 처리 입력 받은 자료를 처리하거나 여러 가지 연산을 처리할 때 사용한다.
판단		문제의 조건 조건이 참일 경우는 '예(True/Yes)', 조건이 거짓일 경우에는 '아니오(False/No)'로 표시, 조건에 따라 처리할 때 사용한다.
출력		출력 처리 결과를 서류나 파일로 출력할 때 사용한다.
흐름선		실행 순서 실행 순서로, 기호를 연결할 때 사용한다.

〈작성 요령〉

- 순서도를 작성할 때는 모든 사람이 이해하기 쉽게 약속된 기호를 사용한다.
- 순서도 기호에 처리 내용을 한글, 영문 또는 수식의 형태로 기술하며, 필요하면 기호의 외부에 주석을 붙여 그 내용을 명확히 한다.
- 처리 순서에 따라 흐름이 위에서 아래로, 왼쪽에서 오른쪽으로 흘러가도록 작성하며, 그렇지 않은 흐름은 화살표로 표시한다.
- 한 페이지에 작성할 수 없거나 어떤 조건에 의하여 분기가 이루어질 때는 연결 기호를 사용하여 표시한다.
- 시작과 끝 처리의 블록에는 단말 기호를 이용하여 표시하고 반복 처리의 내용을 명확히 한다.

순서도는 흐름에 따라 다음과 같이 세 가지 유형으로 나눌 수 있다.

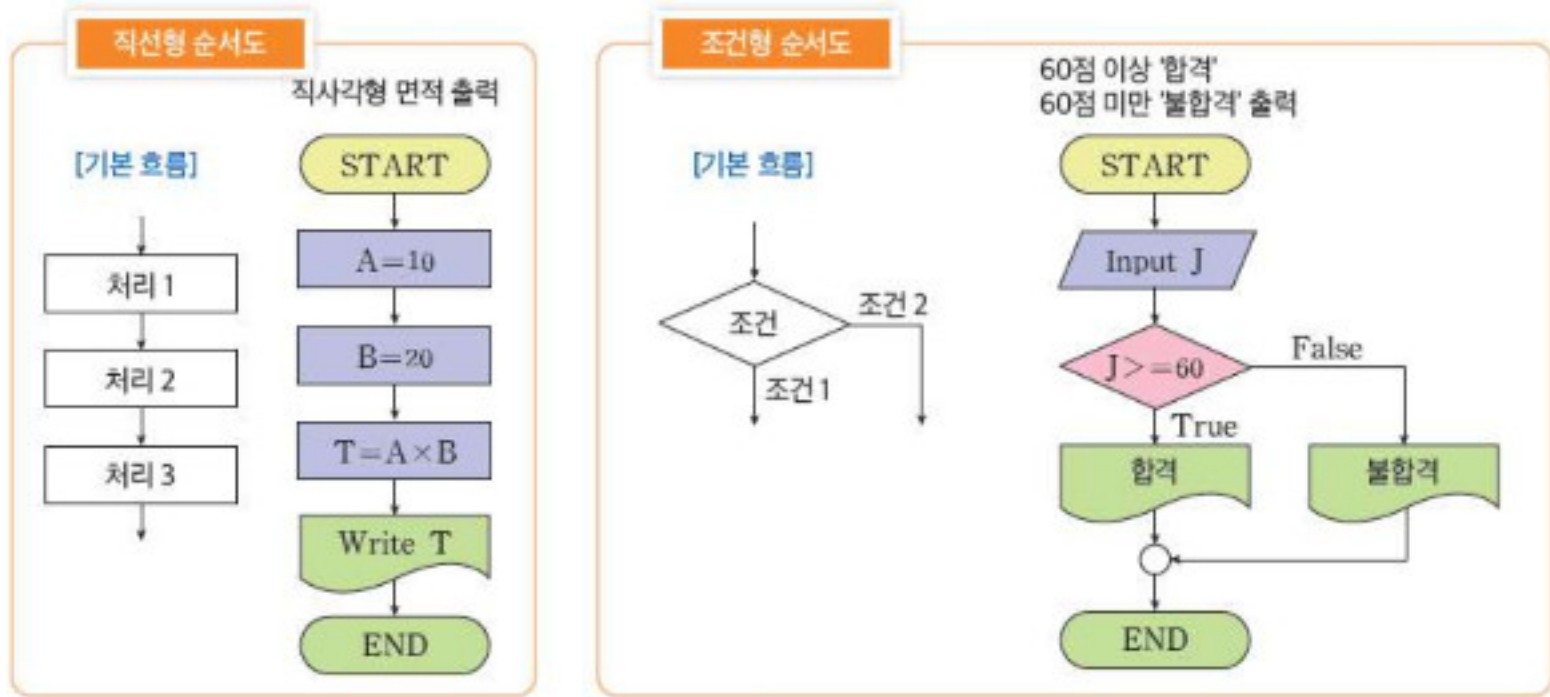


그림 V-18 순서도의 종류

작업 따라잡기

각각의 순서도를 작성해 보자.

(1) 4개의 숫자를 입력받아 합과 평균을 구하는 순서도를 작성해 보자.

구분	내용
순서도 유형	
알고리즘 구조	
입력	
출력	

순서도 작성

(2) 1부터 100까지 합을 구하는 순서도를 작성해 보자.

구분	내용
순서도 유형	
알고리즘 구조	
입력	
출력	

순서도 작성

(3) 두 개의 숫자를 비교하여 큰 수를 출력하는 순서도를 작성해 보자.

구분	내용
순서도 유형	
알고리즘 구조	
입력	
출력	

순서도 작성



문제

화씨 온도를 섭씨 온도로 변환하기 위한 프로그램을 순서도로 작성해 보자.

1 문제 파악 및 정의

문제를 파악하고 정의한다.
화씨 온도를 섭씨 온도로 변환한다.

2 알고리즘 작성

알고리즘을 프로그래밍 언어로 변환한다.

화씨 온도를 입력한다.



변환식 : $(\text{화씨 온도} - 32) \times 9/5$

화씨 온도를 섭씨 온도로
변환하는 식에 대입한다.



섭씨 온도를 출력한다.

구분	내용
순서도 유형	
알고리즘 구조	
입력	
출력	

순서도 작성



문제

1. 키보드로 임의의 양의 정수 N 을 입력받아 1부터 N 까지의 짝수의 합을 구하는 순서도를 작성하시오.
(예: $N=10$, 1부터 10까지 짝수의 합은 30)

순서도 작성

2. 소수란 1과 자기 자신만으로 나뉘는 1보다 큰 양의 정수를 의미한다. 키보드로 임의의 정수 N 을 입력받아 소수를 판별하는 순서도를 작성하시오.

순서도 작성

01. 문제 해결 과정

(1) 문제

문제는 해결해야 할 과제나 상황을 의미하며, 문제를 해결한다는 것은 원하는 결과를 얻어낸다는 것

(2) 문제 해결 과정

01 단계

문제 이해 및 분석하기

문제를 분석하여 도식화하고 구조화하여 문제 해결 방법을 찾는 데 활용한다.

02 단계

문제 해결 방법 찾기

거꾸로 해결하기, 작은 문제부터 해결하기, 여러 번의 시도와 실수를 통해 해결하기 등의 다양한 방법을 이용한다.

03 단계

문제 해결하기

- 2단계에서 선택한 방법을 이용하여 문제를 해결한다.
- 컴퓨터를 이용하여 문제를 해결하는 경우에는 문제 해결 방법을 프로그램으로 구현한다.

04 단계

문제 해결 방법에 대해 평가하기

문제 해결 시간, 기억 장소의 사용량 등을 검사하여 문제 해결 방법을 평가 후 필요할 때는 1단계나 2단계로 되돌아가서 다른 해결 방법을 찾아 문제를 해결한다.

(3) 문제 표현 방법

① 글로 표현하기

머릿속에 있는 문제를 글로 표현해 보는 것

② 표로 표현하기

핵심적인 내용을 일목요연하게 정리하여 표의 형태로 표현하는 것

③ 그림과 기호로 표현하기

약속된 기호나 도형을 이용하여 제시된 문제를 시각화하는 것

(4) 컴퓨터와 문제 해결

① 문제 이해 및 분석

해결하려고 하는 것이 무엇인지 찾기 위해 제시된 문제를 잘 이해하고 분석하는 단계

② 문제 해결 방법 찾기

일반적인 문제 해결 과정의 단계와 동일하게 주어진 문제를 해결하기 위한 가장 효율적인 방법을 찾아보는 단계

③ 알고리즘 설계

알고리즘 설계는 문제 해결 절차나 방법을 알기 쉽게 기술하는 단계

④ 프로그램 작성

컴퓨터를 이용한 문제 해결 과정에서는 컴퓨터에서 실행할 수 있도록 알고리즘 설계를 바탕으로 프로그래밍 언어를 이용하여 프로그램을 작성해야 하는 부분이 있는지 확인하는 단계

(5) 추상화(abstraction)

추상화는 문제 해결을 위해 문제를 분해하거나 중요한 부분을 끌어내는 것 등을 통하여 해결해야 할 문제를 보다 해결하기 쉬운 형태로 변환하는 과정

(6) 문제의 분해(problem decomposition)

문제의 분해는 주어진 문제를 여러 개의 작은 문제들의 집합으로 볼 수 있는 경우에 적용하는 문제 이해 및 분석 방법

(7) 모델링(modeling)

모델링이란 문제 해결을 위해 문제 내부에 포함된 구성 요소들 간의 관계를 단순화해 표현하고 구조화하는 것



대단원 평가문제

01 문제 해결 과정 중 ㉠에 해당하는 것은?

1단계	㉠	문제를 분석하여 도식화하고 구조화하여 문제 해결 방법을 찾는 데 활용한다.
↓		
2단계	㉡	거꾸로 해결하기, 작은 문제부터 해결하기, 여러 번의 시도와 실수를 통해 해결하기 등의 다양한 방법을 이용한다.
↓		
3단계	㉢	2단계에서 선택한 방법을 이용하여 문제를 해결한다. - 컴퓨터를 이용하여 문제를 해결하는 경우에는 문제 해결 방법을 프로그램으로 구현한다.
↓		
4단계	㉣	문제 해결 시간, 기억 장소의 사용량 등을 검사하여 문제 해결 방법을 평가 후 필요할 때는 1단계나 2단계로 되돌아가서 다른 해결 방법을 찾아 문제를 해결한다.

- ① 문제 이해하기
- ② 문제 만들기
- ③ 문제 해결하기
- ④ 문제 해결 방법 찾기
- ⑤ 문제 해결 방법에 대해 평가하기

02 빈칸에 알맞은 문제 해결 과정으로 적합한 것은?



- ① 알고리즘 설계
- ② 문제 이해 및 분석
- ③ 문제에 재적용
- ④ 해결 과정 분석
- ⑤ 프로그래밍 코딩

03 문제를 그림과 기호로 표현했을 때 얻을 수 있는 장점이 아닌 것은?

- ① 문제를 한눈에 쉽게 파악할 수 있다.
- ② 기호가 사전에 약속되어 있지 않아도 이해하기 쉽다.
- ③ 복잡한 문제가 간단하게 도식화된다.
- ④ 자료가 시각화되어 좋다.
- ⑤ 알기 쉽게 정보를 전달해 줄 수 있다.

04 다음이 설명하는 것은 무엇인가?

문제 해결을 위해 문제를 분해하거나 중요한 부분을 끌어내는 것 등을 통하여 해결해야 할 문제를 보다 해결하기 쉬운 형태로 변환하는 과정

05 다음 문제 표현 방법에 대한 내용을 채우시오.

문제 표현 방법	글로 표현
	표로 표현

06 컴퓨터로 문제 해결을 표현하기 위한 과정을 기술하고, 그 중 하나를 선택하여 특징을 서술하시오.

특징

과정명	원리 또는 특징